



AF  
\$ 2700

Please type a plus sign (+) inside this box → ☐

PTO/SB/21 (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

<b>TRANSMITTAL FORM</b> <i>(to be used for all correspondence after initial filing)</i>	<b>Application Number</b>	09/294,656	
	<b>Filing Date</b>	04/19/1999	
	<b>First Named Inventor</b>	M. Cusson, et al.	
	<b>Group Art Unit</b>	2172	
	<b>Examiner Name</b>	Colbert	
<b>Total Number of Pages in This Submission</b>	104	<b>Attorney Docket Number</b>	oracle01.001

ENCLOSURES (check all that apply)		
<input checked="" type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Assignment Papers (for an Application)	<input type="checkbox"/> After Allowance Communication to Group
<input checked="" type="checkbox"/> Fee Attached	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
<input type="checkbox"/> Amendment / Reply	<input type="checkbox"/> Licensing-related Papers	<input checked="" type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
<input type="checkbox"/> After Final	<input type="checkbox"/> Petition	<input type="checkbox"/> Proprietary Information
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Petition to Convert to a Provisional Application	<input type="checkbox"/> Status Letter
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address	<input type="checkbox"/> Other Enclosures (please identify below)
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Terminal Disclaimer	<input type="checkbox"/> Return Postcard
<input type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> Request for Refund	
<input type="checkbox"/> Certified Copy of Priority Document(s)	<input type="checkbox"/> CD, Number of CD(s) _____	
<input type="checkbox"/> Response to Missing Parts/Incomplete Application	<b>Remarks</b>	
<input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53		

RECEIVED  
MAR 06 2003  
GROUP 3600

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Gordon E. Nelson, #30,093
Signature	<i>Gordon E. Nelson</i>
Date	02/27/2003

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: 2/27/03			
Typed or printed name	Gordon E. Nelson, #30,093		
Signature	<i>Gordon E. Nelson</i>	Date	2/27/03

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



# FEE TRANSMITTAL for FY 2002

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT (\$ ) 320.00

## Complete if Known

Application Number	09/294,656
Filing Date	04/19/1999
First Named Inventor	M. Cusson, et al.
Examiner Name	Colbert
Group Art Unit	2172
Attorney Docket No.	oracle01.001

## METHOD OF PAYMENT

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number: 501315  
Deposit Account Name: Gordon E. Nelson

☐ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Applicant claims small entity status. See 37 CFR 1.27

2. ☒ Payment Enclosed:

☒ Check ☐ Credit card ☐ Money Order ☐ Other

## FEE CALCULATION

### 1. BASIC FILING FEE

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	

SUBTOTAL (1) (\$ )

### 2. EXTRA CLAIM FEES

Total Claims:  -20\*\* =  X  =   
Independent Claims:  -3\*\* =  X  =   
Multiple Dependent:  =

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$ )

\*\*or number previously paid, if greater; For Reissues, see above

## FEE CALCULATION (continued)

### 3. ADDITIONAL FEES

Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description
105	130	205	65	Surcharge - late filing fee or oath
127	50	227	25	Surcharge - late provisional filing fee or cover sheet
139	130	139	130	Non-English specification
147	2,520	147	2,520	For filing a request for <i>ex parte</i> reexamination
112	920*	112	920*	Requesting publication of SIR prior to Examiner action
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action
115	110	215	55	Extension for reply within first month
116	400	216	200	Extension for reply within second month
117	920	217	460	Extension for reply within third month
118	1,440	218	720	Extension for reply within fourth month
128	1,960	228	980	Extension for reply within fifth month
119	320	219	160	Notice of Appeal
120	320	220	160	Filing a brief in support of an appeal
121	280	221	140	Request for oral hearing
138	1,510	138	1,510	Petition to institute a public use proceeding
140	110	240	55	Petition to revive - unavoidable
141	1,280	241	640	Petition to revive - unintentional
142	1,280	242	640	Utility issue fee (or reissue)
143	460	243	230	Design issue fee
144	620	244	310	Plant issue fee
122	130	122	130	Petitions to the Commissioner
123	50	123	50	Processing fee under 37 CFR 1.17(q)
126	180	126	180	Submission of Information Disclosure Stmt
581	40	581	40	Recording each patent assignment per property (times number of properties)
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))
149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))
179	740	279	370	Request for Continued Examination (RCE)
169	900	169	900	Request for expedited examination of a design application

Other fee (specify) \_\_\_\_\_

\*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$ ) 320.00

## SUBMITTED BY

Name (Print/Type) Gordon E. Nelson

Registration No. 30,093  
(Attorney/Agent)

Signature *Gordon E. Nelson*

## Complete (if applicable)

Telephone 978-948-7632

Date 02/27/2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



Oracle01.001

ORACLE CONFIDENTIAL

**PATENT**  
**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
(Oracle01.001)

#23  
m  
3/11/03

5   **Applicant:**   M. Cusson, et al.

**Paper No.:** 20

**Application No:**   09/294,656

**Group Art Unit:** 2172

**Filed:**           4/19/99

**Examiner:** Colbert

10

**Title:**           *Web servers with queryable dynamic caches*

.....  
Assistant Commissioner for Patents  
15   Washington, DC 20231

**Appeal Brief under 37 C.F.R. 1.192**

RECEIVED  
MAR 08 2003  
GROUP 3600

20   **(1) Real party in interest**

Oracle International Corporation, a California corporation having an address at 500 Oracle Parkway, Redwood Shores, CA 94065.

25   **(2) Related appeals and interferences**

None

**(3) Status of claims**

30   The claims presently in the application are claims 24-35, 53-60, 84-93, and 112-131. Claims 112-131 have been rejected under 35 U.S.C. 112, first paragraph as lacking support in the original Specification. Claims 24-35, 53-60, and 84-93 have all been rejected under 35 U.S.C. 103(a) as unpatentable over Draper, et al., U.S. Patent 5,924,096, henceforth "Draper".

35   **(4) Status of amendment**

The claims have not been amended since claims 24-35, 53-60, and 84-93 were elected in a response filed 3/12/2002 to a restriction requirement made on 2/12/2002. In that same response, Applicants added new claims 112-131, which also have not been amended.

03/03/2003 SZEWIDIE1 00000087 09294656

01 FC:1402

40

**(5) Summary of the inventions**

320.00 OP

*The inventions of claims 24-35, 53-60, and 84-93*

Claims 24-35, 53-60, and 84-93 are closely related to each other. All of them involve a "server of the type that provides a program with a standard interface for querying remote datasets" (claim 24, lines 1-2). Claims 24-35 were in the application as filed, but independent claims 24 and 34 were amended to broaden them in Applicants' response filed 10/20/00 to a first Office action in the application. Claims 24-33 are apparatus claims directed to an "improved" server of the type, claims 34-35 and 53-60 are method claims directed to methods practiced in a server of that type, and claims 84-93 are Beauregard claims to a memory device which contains code that when executed by a processor, performs the methods of claims 34-35 and 53-60. Given the close relationships between the claims, Applicants believe it is necessary only to summarize claims 24-33 in detail.

Claim 24

Independent claim 24 reads as follows:

24. (twice amended) An improved server of the type that provides a program with a standard interface for querying remote datasets,

the improved server comprising:

a queryable cache that contains copies of certain of the datasets and is local to the server,

the improved server receiving a query for a remote dataset in a form required by the interface from the program, determining whether a copy of a dataset to be queried is present in the queryable cache, and, if the copy is present, querying the copy, and otherwise querying the remote dataset,

whereby the queryable cache is transparent to the program.

The preamble's prior art "server of the type that provides a program (Web application 111) with a standard interface for querying remote datasets" is disclosed at 107 in Applicants' FIG. 1 and described at page 1, line 31 through page 2, line 24.

A problem of system 101 is that when data is cached in server 107, it is cached in the form of Web pages, and consequently is not *queryable*. The Specification as filed defines *queryable* negatively at page 5, lines 15-18 as follows:

5       since the data is cached in the form of HTML pages, it is not in *queryable* form, that is, a cached HTML page may contain data from which another query received from Web browser 103 could be answered, but because the data is contained in an HTML page instead of a database table, it is not in a form to which a query can be applied.

10      This problem is solved by the inventions set forth in claims 24-35, 53-60, and 84-93. The solution provided by the inventions is set forth in broad terms at page 6, lines 8-11 of Applicants' Specification:

15       The problem of queryable data is solved by using a database system in the server as the cache. If the information necessary to run the query is present in the cache database system, the query is run on the cache database system; otherwise, it is run on a source database system.

20      It will be immediately apparent to those skilled in the relevant arts that the term *query* is being used here and throughout Applicants' Specification in its database sense, that is, a query is expressed in a query language and the query as expressed in the query language is then executed in a database system that can interpret the query language. In the preferred embodiment, the query language is SQL and the database system is a relational database system.

25      FIG. 2 provides a high-level overview of an embodiment of the invention. An improved server 203(i) is shown which is like prior art server 107 except that it includes a queryable cache 219. Queryable cache 219 is a database that has a copy 223 of a portion of the data in source database 241 of source DB server 237. Data access layer 253 of improved server 203 permits improved server 203 to access either cache database 236 (erroneously 226 in FIG. 2) or source  
30      database 241 in source DB server 237. Operation of server 203 with queryable cache 219 is described at page 8, lines 5-19:

35       When data access 253 receives a query from web application 111, it first presents the query to queryable cache 219, as shown at Q 215. If cached data 223 includes the data specified in the query, queryable cache 219 returns result (R) 217, which data access 253 returns to Web application 111. If cached data 223 does not include the data specified in the query, queryable cache 219 returns a

miss signal (M) 216 to data access 253, which then makes the query via network 113 to source database server 237 and when it receives the result, returns it to Web application 111. The query made in response to the miss signal appears as miss query (MQ) 224 and the response appears as miss response (MR) 226.

It is important to note here that because the interactions with queryable cache 219 and with source database server 237 are both performed by data access layer 253, the existence of queryable cache 219 is completely *transparent* to Web application 111. That is, a Web application program 111 that runs on Web server 107 will run without changes on Web server 203(i).

#### Claims 25 and 26

Claim 25 provides details of a species of claim 24. The claim adds the limitations that the program uses global identifiers to identify the remote data sets, that the copies in the queryable cache have local identifiers, and that the improved server further comprises

a query analyzer that receives the global identifier for a dataset being queried and if there is a copy of the data set indicated by the global identifier, returns the local identifier to the server,

the server using the local identifier to query the copy. (claim 25, lines 5-7)

This species of claim 24 is shown in detail in FIG. 3. The global and local identifiers are described at page 12, lines 2-13. The query analyzer of the claim appears at 313 in FIG. 3 and its operation is described at page 12, lines 17-31 and shown at 607, 609, 611, 613, and 615 of flowchart 601 in FIG. 6.

Claim 26 adds the limitation to claim 24 that the query analyzer further indicates to the server whether the copy of the dataset is in the queryable cache. The disclosure which supports this claim may be found at the same locations as the disclosure for claim 25.

#### Claims 27-33

These claims are all directed to various techniques for keeping the datasets in cache database 236 current. The problem addressed by these techniques is set forth at page 4, line 33 through page 5, line 11. The solutions set forth in these claims are summarized at page 5, line 31 through page 6, line 2.

Continuing with the claims in detail, the dataset manager that appears as a new limitation in claim 27 is shown at 213 in FIG. 2 and at 323 in FIG. 3. Its operation is explained at page 9, line 31 through page 11, line 10 with regard to FIG. 2 and at page 14, line 28 through page 15, line 2. These same locations support the further limitations concerning the dataset manager that are set forth in claims 28, 29, 31, and 32. The query log of claim 30 is a part of query information 208, as explained at page 10, lines 12-17, which also explain how the dataset manager uses the query log to determine whether to add or remove a dataset from cache database 236.

In claim 33, finally, the added limitation of an "update receiver" is shown at 210 in FIG. 2 and 321 in FIG. 3. How the source DB server and the update receiver interact when changes to cached datasets are made in source database 241 is explained at page 9, lines 21 through 30 and at page 13, lines 21-26.

Method claims 34-35 and 53-60 and Beauregard method claims 84-93

The scopes of these claims are of course not identical to those of claims 24-33, but their relationships to Applicants' Specification are the same; consequently, Applicants are providing the following table, which relates claims 24-33 to claims 34-35 and 53-60 and to claims 84-83.

Claims 24-33	Claims 34-35 and 53-60	Claims 84-93
24	34	84
25	35	85
26	53	93
27	54	86
28	55	87
29	56	88
30	57	89
31	58	90
32	59	91
33	60	92

*The inventions of claims 112-130*

These claims were added to the application in an amendment filed 3/12/2002. Claims 112-118 are addressed to "Apparatus for responding to a request"; claims 119-123 are addressed to "A method of responding to a request"; claim 124 is a Beauregard claim which incorporates the method steps of claim 119; claims 125-127 are directed to "Apparatus for caching copies of objects belonging to a subset of the objects belonging to a first database system; claims 128-130 are directed to "A method of responding to a request" whose steps are performed in a context like that provided by the invention of claim 125; claim 131, finally, is a Beauregard claim which incorporates the method steps of claim 128.

Claims 112-131 address the system disclosed in the patent application under appeal as "a distributed database system that includes a plurality of database systems" (claim 112, lines 2-3) and as "apparatus for caching copies of objects" from a first database system in a second database system (claim 125, lines 1-5). In the following, a summary of claims 112-131 will be presented; the issue of whether claims 112-131 are supported by the specification as filed will be discussed in the section *Argument* below.

Claims 112-124

Claims 112-124 are sufficiently closely related that a detailed summary of apparatus claims 112-118 should suffice for all of them. Claim 112 reads as follows:

112. Apparatus for responding to a request, the request including one or more specifiers referring to objects belonging to a plurality thereof in a distributed database system that includes a plurality of database systems and the apparatus comprising:  
a first database system of the plurality; and  
a redirector which responds to the request when the request includes a specifier that cannot be interpreted in the first database system by causing the request to be executed at least in part in a second database system of the plurality, the request otherwise being executed in the first database system.

Beginning with the preamble, the preamble contains the terms "request", "specifiers", "objects", "distributed database system". FIG. 2 shows the "distributed database system that includes a



plurality of database systems" disclosed in the patent application; it is embodied in a cache database system 347 in each of the servers 203 and a source database system 241 in source database server 237. The "objects" in the distributed database system are embodied in database objects such as database tables (see page 8, line 23 of Applicants' Specification). The

5 "request" is embodied in the OLE-DB, ODBC, or JOBC queries which data access layer 253 received from Web application 111, as explained at page 2, lines 16-24; the "identifiers" are embodied in the portions of the queries that identify database objects such as the tables.

Continuing with the body of the claim, the "first database system" is embodied in cache  
10 database 347; the "second database system" is embodied in source database 241. The "redirector" is embodied in DA interface 304 and query dispatcher 351. The behavior of the redirector set forth in lines 6-9 of claim 112 is described at page 12, line 15 through page 13, line 30.

15 Continuing with claim 113, the added limitation is embodied in the fact that cache database 347 contains copies of objects from source database 241. Claim 114 expressly sets forth the limitation that the database system embodied in cache database 347 is a cache. Claims 115 and 116 address the relationship between cache database 347 and source database 237 embodied in system 201. Claims 117 and 118 are addressed to embodiments like system 201 in which the  
20 "Apparatus for responding to a request" is local to a Web server of the type shown at 107 in FIG. 1. Method claims 119-123 are different in scope from apparatus claims 112-118, but are embodied in the same features and behavior of system 201, as is Beauregard claim 124, which is based upon method claim 119.

25 Claims 125-131

These claims are also sufficiently closely related that a detailed summary of claims 125-127 should suffice for all of them. Claim 125 reads as follows:

30 **125.** Apparatus for caching copies of objects belonging to a subset of the objects belonging to a first database system that returns an object in response to a request therefor, the request including one or more specifiers referring to the objects and the apparatus comprising:

a second database system that contains the copies; and

a redirector that responds to the request when the request includes a specifier that cannot be interpreted in the second database system by causing the request to be executed at least in part in the first database system, the request otherwise being executed in the second database system.

5

Beginning with the preamble, the "first database system" is embodied in source database 241; "objects" are embodied as explained above with regard to claim 112, as are "requests" and "specifiers referring to the objects". The "second database system" is embodied in cache database 347 and the redirector is embodied in query dispatcher 351 and DA interface 304, as explained in the discussion of claim 112 above. Claims 126 and 127 are embodied as described for claims 122 and 123 above. Method claims 128-130 have different scopes from apparatus claims 125-127, but are embodied in the same features and behavior of system 201, as is claim 131, which is a Beauregard claim based on method claim 128.

#### 15 (6) Issues

The issues are the following:

1. Does Applicants' Specification as filed satisfy the "written description" requirement of 35 U.S.C. 112 with regard to claims 112-131, or to put it in more traditional terms, are Applicants' claims 112-131 supported by Applicants' Specification as filed?
- 20 2. Are claims 24-35, 53-60, and 84-93 unpatentable under 35 U.S.C. 103 as obvious over U.S. patent 5,924,096, Draper, et al.?

#### (7) Grouping of claims

*Claims 112-131*

25 As set forth in detail in the *Summary of the invention, supra*, claims 112-118 are apparatus claims, claims 119-123 are method claims, and claim 124 is a Beauregard apparatus claim. All are addressed to substantially the same inventive concepts, so the claims stand and fall together as set forth in the following table:

Group number	Claims in group
11	112,119,124

12	113,120
13	114,--
14	115,121
15	116,--
16	117,122
17	118,123

The same is true with regard to claims 125-131. Claims 125-127 are apparatus claims, 128-130 are method claims, and claim 131 is a Beauregard method claim. All are addressed to substantially the same inventive concepts, so the claims stand and fall together as follows:

Group number	Claims in group
18	125,128,131
19	126,129
20	127,130

5

*Claims 24-35, 53-60, and 84-93*

10 As also set forth in detail in the *Summary of the Invention, supra*, claims 24-33 are apparatus claims, claims 34-35 and 53-60 are method claims, and 84-93 are Beauregard method claims that are addressed to substantially the same inventive concepts. Corresponding claims of each of the three classes of course have different scopes, but will stand and fall together with regard to the rejections under 35 U.S.C. 103. The groups of claims that stand and fall together are as follows:

15

Group number	Claims in group
1	24,34,84
2	25,35,85
3	26,53,93
4	27,54,86
5	28,55, 87

6	29,56,88
7	30,57,89
8	31,58,90
9	32,59,91
10	33,60,92

### (8) Argument

The following *Argument* will first address the rejections of claims 112-131 under 35 U.S.C. 112, first paragraph and then the rejections of the remaining claims as obvious over Draper.

5

*The rejection of claims 112-131 under 35 U.S.C. 112, first paragraph*

The reasons for of Examiner's rejection of claims 112-131 under 35 U.S.C. 112, first paragraph are clearly set forth at page 2 of her Final Office Action of 12/2/02:

10 Newly added claims 112-131 are rejected under 35 U.S.C. 112, first paragraph as containing subject matter which was not described in the specification. The added material which is not supported by the original Specification is as follows: Claims 112-131 recite "objects, distributed database system, redirector, or specifier/specifiers" which are not mentioned in Applicants' Specification or drawings. Therefore, there is a lack of support in Applicants' Specification for  
15 these claim limitations.

What Examiner is requiring here is that newly-added claims use only terms that appear expressly in the Specification as filed. As Applicants' attorney pointed out in his response to the non-final Office action of 6/4/02 in which the above rejection first appeared, the first paragraph  
20 of 35 U.S.C. 112 makes no such requirement. See in this regard the discussion at MPEP 2163.02, which states:

25 The subject matter of the claim need not be described literally (i.e., using the same terms or *in haec verba*) in order for the disclosure to satisfy the description requirement. (MPEP 2100-167, col.2, August 2001)

What the first paragraph of 35 U.S.C. 112 *does* require with regard to the "written description" of the invention has been set out in detail in a recent case of the Federal Circuit, *New Railhead Mfg. L.L.C. v. Vermeer Mfg. Co.*, 298 F.,3d 1290 at 1296, (Fed. Cir. 2002):

30 The purpose of the written description requirement is broader than to merely explain how to 'make and use'; the applicant must also convey with reasonable

clarity to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention." Vas-Cath Inc. v. Mahurkar, 935 F.2d 1555, 1563-64, 19 U.S.P.Q.2D (BNA) 1111, 1117 (Fed. Cir. 1991). That is, the disclosure must show he had invented each feature that is included as a claim limitation. The adequacy of the written description (i.e., the disclosure) is measured from the face of the application; the requirement is not satisfied if one of ordinary skill in the art must first make the patented invention before he can ascertain the claimed features of that invention. Cf. Martin v. Mayer, 823 F.2d 500, 505, 3 U.S.P.Q.2D (BNA) 1333, 1337 (Fed. Cir. 1987) ("It is not a question of whether one skilled in the art might be able to construct the patentee's device [\*\*14] from the teachings of the disclosure [but] whether the application necessarily discloses that particular device." (quoting Jepson v. Coleman, 50 C.C.P.A. 1051, 314 F.2d 533, 536, 136 U.S.P.Q. (BNA) 647, 649-50 (CCPA 1963))).

In the claims originally filed in the patent application, Applicants claimed their invention in the context of one of the most interesting applications for it, namely in the context of a network server. What Applicants are doing in their claims 112-131 is claiming the invention in the context of database systems generally, not just database systems that are used with network servers. As Applicants will demonstrate in the following, the disclosure of the invention in the context of the network server is also *necessarily* a disclosure of the invention in the context of database systems generally, and for that reason, the disclosure as filed completely satisfies the written description requirement with regard to claims 112-131.

Beginning with "objects", that is simply a generic term for the things that are stored in a database system. Such a use of the term is common in the database arts, as may be seen from Leverenz, et al., *Oracle8 Server Concepts*, release 8.0, Oracle Corporation, Redwood City, CA, 1998, which is part of the documentation for the Oracle8 servers used in the implementation of the invention that is described in Applicants' Specification. Leverenz, et al., provides a list of *schema objects* at page 1-10 that sheds some light on what is meant by "object" in the database art:

A *schema* is a collection of database objects. *Schema objects* are the logical structures that directly refer to the database's data. Schema objects include such structures as tables, views, sequences, stored procedures, synonyms, indexes, clusters, and database links.

Applicants' Specification speaks of "datasets" stored in queryable cache 219 and source DB 241, but as specifically pointed out at page 8, line 24 of Applicants' Specification, the "datasets"

are database tables. These are clearly "objects" as set forth in the above quotation from Leverenz, and as Oracle8 servers, queryable cache 219 and source DB 219 are well known to contain other kinds of "objects" as well.

5 As for "distributed database system", Leverenz, et al. defines that term as follows:

*A distributed database is a network of databases managed by multiple database servers that appears to a user as a single logical database. (Leverenz, 1-24)*

10 By the above definition, Applicants' invention as disclosed in FIGs. 2 and 3 is clearly a "distributed database system". There can be any number of servers with queryable caches 203, each of which contains a database 236, and a source DB server 237, and because queryable cache 219 is *transparent* to Web application 111, the databases of system 201 appear to the user as a single logical database. (See in this regard p.8, lines 15-19.)

15

The meaning of "specifier" is defined in claim 112 itself:

one or more specifiers referring to objects belonging to a plurality thereof in a distributed database system

20 In the embodiment described in the Specification as filed, the "specifiers referring to objects" is a query's *context*, which is described at page 9, lines 5 and 6 as being embodied as "a list of the identifiers for the required data sets". Since the datasets are "objects", "a list of the identifiers for the required data sets" is clearly an embodiment of the claim's "specifiers referring to objects belonging to a plurality thereof".

25

The meaning of "redirector" is also defined in claim 112 itself:

a redirector which responds to the request when the request includes a specifier that cannot be interpreted in the first database system by causing the request to be executed at least in part in a second database system of the plurality, the request otherwise being executed in the first database system.

30

As Applicants pointed out when they added claims 112, 131 to the application,

The "redirector" is embodied in DA interface 304 and query dispatcher 351. The manner in which these components of this embodiment of the invention interact to "respond[]" to the request when the request includes a specifier that cannot be

35

interpreted in the first database system by causing the request to be executed at least in part in a second database system of the plurality, the request otherwise being executed in the first database system" is described at least at page 12, lines 15-24. (Applicants' response of 3/12/02, p.16, lines 18-23)

The relevant portion of the Specification as filed, page 12, line 15, through page 13, line 7, reads as follows:

Data access layer 349 includes a new component, query dispatcher 351, which is the interface between data access layer 349 and queryable cache 302. FIG. 6 is a flowchart 601 of the operation of query dispatcher 351 in a preferred embodiment. Reference numbers in parentheses refer to elements of the flowchart. When data access layer 349 is preparing to query source database 241, it provides the global context for the query to query dispatcher 351 (605), which in turn provides global context 318 (FIG. 3) to query analyzer 313 (607). Query analyzer 313 determines whether the datasets identified by the global context are cached in cache database 347; if they are not, query analyzer 313 reports a miss 319 to query dispatcher 351 (609), which indicates to data access layer 349 that it is to place the global query on network 113.

If the datasets identified by the global context are cached in cache database 347, query analyzer 313 indicates that fact to query dispatcher 351 and also provides query dispatcher 351 with local context 316 for the datasets in cache database 347 (615). Query dispatcher 351 then provides the local context to data access layer 349, which uses the local context to make a local query 317 corresponding to the global query and then uses the local query to obtain local result 320 from cache database 347. It should be noted here that the operations involved in the translation from the global query to the local query and applying the local query to cache database 347 may be divided among data access layer 349, query dispatcher 351, and query analyzer 313 in many different ways; the advantage of the technique of flowchart 601 is that data access layer 349 can employ the same mechanisms to make local queries as it does to make global queries. All query analyzer 313 and query dispatcher 351 need do is supply data access layer 349 with the local context needed to make the local query.

Clearly, between them, query dispatcher 351 and data access interface 304 perform the operation that defines the redirector in claim 112. When the request includes a specifier (here, the global context received from the application program), query dispatcher 351 refers it to query analyzer 313 in DA interface 304 to determine whether the specifier specifies datasets that are present in cache database 347; if it does not, the specifier "cannot be interpreted in the first database system" (cache database 347) and query dispatcher 351 causes the request to be executed instead "in a second database system of the plurality", namely source database server 237. If the specifier "can be interpreted in the first database system", query dispatcher 351

executes the query there. Query dispatcher 351 and data access interface 304 are thus an embodiment of the claim's "redirector".

Since the claim limitations that Examiner believed were not supported by the Specification are in fact supported by it, claim 112 satisfies the requirements of the written description requirement as set forth in *New Railhead, supra*. The arguments just made apply equally to independent claims 119 and 131 and to the claims dependent from claims 112 and 119.

Examiner does not expressly identify any reason why claims 125, 128, and 131 are lacking support. "objects" and the "redirector" are supported in the same fashion as the same terms in claims 112 and 119; the "first database system" is embodied in source database server 237 and the "second database system" is embodied in cache database 347. Thus, these claims and the claims dependent from claim 125 and 128 are fully supported by the Specification and the claims therefore satisfy the requirements of the written description requirement. Since that is so, Examiner's rejection of claims 112-131 under 35 U.S.C. 112, first paragraph is without foundation

*The rejections of claims 24-35, 53-60, 84-93 as unpatentable over Draper under 35 U.S.C. 103*

This rejection has been in the prosecution since Examiner's Office action of 6/4/02. Applicants traversed the rejection in their response of 9/3/02. It should also be pointed out that claims 24 and 34 were rejected under 35 U.S.C. 102(e) as anticipated by this reference in the Office action mailed 6/20/00 in the application and that Applicants successfully traversed that rejection in their response mailed 10/20/00 in the application.

Applicants are traversing this rejection on two grounds:

1. A procedural ground: The rejection is not made in the manner required for a rejection under 35 U.S.C. 103; and
2. A substantive ground: The reference does not disclose what Applicant is claiming and consequently cannot serve as a basis for the rejection of Applicants' claims under either 35 U.S.C. 102 or 35 U.S.C. 103.



The procedural ground

As set forth in MPEP 2142, in order to make a *prima facie* rejection under 35 U.S.C. 103, the examiner must meet three basic criteria:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (MPEP 2100-121, 2001)

In her rejection of claim 24 under 35 U.S.C. 103, Examiner admits that Draper does not disclose the limitations of "determining whether a copy of the dataset to be queried is present in the queryable cache and if the cache is present, querying the copy and otherwise querying the remote dataset." (Final Office Action page 3, §7). Once having admitted that, the methodology of MPEP 2142 *requires* that Examiner must find other references or use scientific knowledge or official notice or common knowledge to supply the missing limitations. Only *after* she has provided sources that teach or suggest *all* of the claim limitations can she proceed to the next step, which is explaining why one of ordinary skill in the art would be motivated to modify the reference or combine the teachings to produce what is claimed. Examiner, however, proceeds as follows:

... it would have been obvious to one having ordinary skill in the art at the time the invention was made to determine whether a copy of the dataset to be queried is present in the queryable cache and if the copy is present, querying the copy and otherwise querying the remote dataset . . . and to modify in draper in view of Draper's teachings of querying, copying, and cache because such a modification would allow Draper's system to have a process of extracting data from a database and presenting it for use and using a special memory subsystem which frequently uses data values that are duplicated for quick access. (Final Office Action, page 3, §7)

In proceeding as set forth above, Examiner is telescoping the step of finding references that teach or suggest the limitations and the step of explaining why one of ordinary skill would combine the references. The problem with doing it this way, of course, is that the basis for the rejection is not clear and it is consequently difficult for Applicants to deal with the rejection either by rebutting it or by amending their claims. In order to make the basis clear, Applicants repeat their request in their response of 9/3/02 that Examiner follow the procedure set forth in MPEP 2144.03 and cite a reference that illustrates the theory or shows the common knowledge in the art. Moreover, the limitations for which

Examiner can find no references lie at the very heart of Applicants' invention; that being the case, Applicants feel justified in calling the attention of the Board to the statement in MPEP 2144.03, citing *In re Ahlert*, USPQ 418, 420-421 (CCPA 1970) that

the facts so noticed serve to 'fill the gaps' which might exist in the evidentiary showing and should not comprise the principle evidence upon which a rejection is based.

The substantive ground

Applicants have already recited the requirements for a *prima facie* case of obviousness under 35 U.S.C. 103. In the following, Applicants will show that Draper does not disclose what Examiner believes it does and that Examiner has consequently not made her *prima facie* case.

Claim 24 is exemplary for what Applicants are claiming:

24. (twice amended) An improved server of the type that provides a program with a standard interface for querying remote datasets, the improved server comprising:  
a queryable cache that contains copies of certain of the datasets and is local to the server,  
the improved server receiving a query for a remote dataset in a form required by the interface from the program, determining whether a copy of a dataset to be queried is present in the queryable cache, and, if the copy is present, querying the copy, and otherwise querying the remote dataset,  
whereby the queryable cache is transparent to the program.

The claim is addressed to an improved *server of the type that provides a program with a standard interface for querying remote datasets*. The server comprises *a queryable cache that contains copies of certain of the datasets and is local to the server*. The server further does three things:

1. it *receiv[es]* a query for a remote data set in a form required by the interface from the program;
2. it *determin[es]* whether a copy of the data set to be queried is present in the queryable cache; and
3. *if the copy is present, it quer[ies] the copy and otherwise quer[ies] the remote data set.*

There are many limitations of this claim which are not disclosed in Draper. That is not surprising, since Draper is primarily concerned with techniques employed to maintain consistency between copies of data in distributed database systems. Distributed database systems and caches are described only as examples of situations where Draper's techniques may be applied. Form the point of view of the techniques, it makes no difference whether the copies for which consistency must be maintained are in a cache within a database system or elsewhere or in database systems that are nodes of a distributed database system, and as might be expected by this fact, the disclosure concerning caches and database systems is sketchy at best. Indeed, the very fact that Draper is the best reference that has been found in an application which has gone through 7 Office actions under two examiners is itself telling evidence that the inventions of claims 24-35, 53-60, and 84-93 are in fact novel and non-obvious.

Beginning at a fundamental level, Draper does not disclose "a queryable cache". As indicated above, Draper does disclose database systems with memory caches *within the database system* (col. 1, lines 46-55) and systems which cache HTML pages (col. 9, lines 33-59) As described there, the HTML pages are accessed in the cache by Internet users, which means that they are accessed by URI, not by queries as that term is used in Applicants' Specification. Draper also discloses in FIGs. 1 and 6 an arrangement where the sources of the data are master systems 602 and 604 that are on servers 106 and the caches 608 and 610 that are on clients 110, which are workstations. See in this regard col. 4, lines 23-50, and col. 8, lines 1-21. In the described embodiment of the system of FIG. 6, the caches contain HTML documents that are accessed by URI (col. 9, lines 33-59). There is no indication in Draper that caches 608 and 610 are queryable, and given their locations in workstations, it is doubtful that they would be, since there is no advantage to making a cache at that level of a distributed system queryable. See in that regard page 4, lines 5-7 of Applicants' *Description of the prior art*. Indeed, Draper effectively discloses nothing beyond what is set forth in the general discussion of caching at page 3, line 29-page 4, line 7 of Applicants' *Description of the prior art*.

Nothing of what is disclosed in Draper is a queryable cache as that term is used in claim 24. "*Queryable*" must be understood in the context of Applicants' Specification as having the property that a query can be run on it, where, as already pointed out, the term *query* is being used in its database sense, that is, a query is expressed in a query language and the query as expressed in the query language is then executed in a database system that can interpret the query language. The queryable cache contains "copies of certain of the [remote] datasets", and thus cannot be Draper's memory cache of a database system, which contains copies of datasets that are stored on disk in the database system, and are thus local to the database system. Moreover, anything that is not in the memory cache is retrieved from the database system's disk, not by querying a remote dataset.

The queryable cache also cannot be Draper's caches 608 and 610. As pointed out above, caches 608 and 610 are not disclosed as being queryable, and given their locations, are most probably not queryable. Even if they were, they are not "local to the server", as required by claim 24.

The nodes in the distributed data systems discussed in Draper are of course queryable, but there is nothing in Draper to indicate that any of these nodes is functioning as a cache or is in "[a] ... server of the type that provides a program with a standard interface for querying *remote* datasets" (emphasis added), as required by claim 24.

Continuing with other limitations of claim 24 that are not disclosed in Draper, there is the "standard interface for querying remote datasets". Examiner dismisses the lack of disclosure of such an interface in Draper by stating that "interfaces are well known in the art" at page 8, line 2 of her Final Office action, and Applicants agree in general, but what is set forth in the claim is not an interface generally, but rather "a standard interface for querying remote datasets". The fact that the queryable cache is in a server with such an interface is of course what makes the cache transparent to the user, and as pointed out in the Specification, the transparency of the queryable cache is fundamental to the value of the invention, since programs that work with the standard interface need not be altered to take advantage of the queryable cache.

Examiner herself admits at § 7 of her final Office action that Draper does not disclose the interaction between the server, the queryable cache, and the remote dataset interact set forth at lines 6-9 of claim 24. It should also be noted at this point that the interaction expressly involves the preamble's "standard interface for querying remote datasets", and that that language of the preamble must therefore be treated as a limitation.

Since Draper discloses neither a "queryable cache" as that term is used in claim 24, nor the relationship between the server, the standard interface, and the queryable cache set forth in the claim, nor the manner in which the server employs the queryable cache, Examiner has not made her *prima facie* case with regard to claim 24 and the rejection under 35 U.S.C. 103 is without foundation. As the Board will immediately see, the arguments made above apply equally to claims 34 and 83.

*Rejections of the dependent claims*

As just pointed out, independent claims 24, 24, and 84 are all patentable over Draper; that being the case, all of the claims dependent from these claims are also patentable. It should, however, also be pointed out that the portions of Draper which Examiner relies on for rejecting the dependent claims also do not disclose the added limitations of the dependent claims, and that the dependent claims are consequently patentable in their own rights.

Many of the rejections of the dependent claims are based on the portions of Draper which disclose the indexed tags which Draper uses to keep track of the types and order of operations on the data items in the database system. Tags are defined as follows at col. 5, lines 19-34:

Each data item 202 has an associated tag 204 . . . Each tag 204 value corresponds to an event in the history of the associated data item 202, such as the most recent update to the data item 202. The tags 204 may be designed to allow recovery of earlier versions of the data item 202. The tags 204 may also allow the system to determine which copy of a data item 202 is more recent, so that the most recent data can be propagated during synchronization. . . . Suitable tags 204 include timestamps, version numbers, sequence numbers, update reference numbers, transaction counters, and other means of determining the relative order of operations on data items 202.

As further pointed out at col. 5, lines 41-42, the difference between the tags of Draper and prior-art tags is that Draper's tags are indexed.

- 5 In her rejections of claims 25, 35, and 85, Examiner equates Draper's tags to Applicants' global and local identifiers. The latter are defined as follows at page 11, lines 5-14 of Applicants' Specification:

10 In preferred embodiment 301, Web application 111 uses global data set identifiers in queries. The Web applications 111 in all of the servers 203 use the same set of global data set identifiers. A cache data base 347 in a given server 203 has its own set of local data set identifiers for the data sets cached in cache data base 347. In preferred embodiment 301, then, one may speak of *global* queries and query contexts that use global data set identifiers and *local* queries and query contexts that use local data set  
15 identifiers. In the preferred embodiment, query analyzer 313 uses cached data base description 305 to translate global query contexts into local query contexts.

As is apparent from the foregoing, Applicants' global and local identifiers are used in queries and  
20 have nothing whatever to do with Draper's tags. Consequently, Draper's disclosure of tags cannot render claims 25, 35, and 85 obvious.

Continuing with claims 26, 53, and 93, the further limitation in those claims further specifies that "the query analyzer further indicates to the server whether the copy of the dataset is in the  
25 queryable cache". Examiner cites Draper, col. 7, lines 9-20 as showing this limitation. The cited location is, however, a discussion of event lists. An event list is defined as follows at col. 7, lines 10-13:

Each element in an event list contains enough information to allow a cache  
30 manager to recreate the corresponding event, in order to duplicate the effect of the event and thus update the cache.

Clearly, none of this has anything to do with determining whether a particular copy of the dataset is in the queryable cache, and thus, the additional limitation of claims 26, 53, and 93 is also not disclosed in Draper.

35

With regard to claims 29, 30, 31, 32, 56, 58, 90, and 91, all of these claims set forth various aspects of a data set manager that

determines whether to add or remove a dataset by determining a likelihood that a query will be made to the dataset. (claim 29)

5

There is simply nothing in Draper about a "likelihood that a query will be made" or about using such a likelihood to determine whether to add or remove a dataset. The location cited by Examiner in her rejection, col. 6, lines 15-21, merely discloses how tags are associated with data items. Claim 30 further recites how the data manager may use a query log to determine the  
10 likelihood. In rejecting this claim, Examiner cites col. 6, lines 15-21 again and the description of Draper's update log at col. 10, lines 22-53. The update log is, however, not a log of queries, but a log of information that is needed to synchronize data items in a data base with data items in a cache or a replica of the database.

15 Concerning claims 31, 32, 58, 90, and 91, Examiner confuses "event" as used in Draper with "event" as used by Applicants. In Draper, "each tag value corresponds to an event in the history of the associated data item 202, such as the most recent update of the data item 202." (col. 5, line 22). What is meant by "event" in Applicants' claims is apparent from the following passage from Applicants' Specification:

20 For example, if a company engaged in Web commerce is having a 1-day sale on certain items for which there are datasets in source database 241, query information 208 may indicate the datasets for the items and the time of the 1-day sale. Using that information, DSM 213 can obtain the datasets from source database 241 and cache them in cache database 236 before the beginning of the  
25 sale and remove them from cache database 236 after the end of the sale. (p. 10, lines 5-10)

Clearly, "event" as used in Applicants' claims has only the remotest connection with "event" as used by Draper. Examiner herself admits that Draper does not teach the limitations of claims 33,  
30 60, and 92.

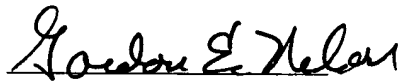
### *Conclusion*

In the foregoing, Applicant has complied with the requirements of 37 C.F.R. 1.192 with regard to his brief and has demonstrated first, that claims 112-131 are fully supported by

Applicants' Specification as filed, and thus satisfy the requirements of 37 C.F.R. 112, first paragraph, and second, that examiner has failed to establish a *prima facie* case of obviousness with regard to *any* of her his rejections of claims 24, 34, and 84 and the claims dependent from these claims under 35 U.S.C. 103. That being the case, the rejections cannot stand and Applicant respectfully requests that the Board of Appeals reverse Examiner with regard to all of her rejections and remand the application to Examiner for further processing as indicated by the reversals.

A check in payment of the fee for filing an appeal brief accompanies this brief; should additional payment be necessary, please charge it to deposit account 501315; any overpayment should also be credited to that account.

Respectfully submitted,



Attorney of record,  
Gordon E. Nelson  
57 Central St., P.O. Box 782  
Rowley, MA, 01969,  
Registration number 30,093  
Voice: (978) 948-7632  
Fax: (866) 723-0359

2/27/03  
Date

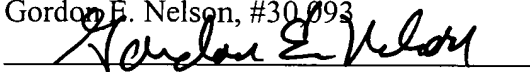
### Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:

Assistant Commissioner for Patents  
Washington, D.C. 20231

on 2/27/03  
(Date)

Gordon E. Nelson, #30,093





**(9) Appendix of claims**

1     **24.** (twice amended) An improved server of the type that provides a program with a  
2     standard interface for querying remote datasets,  
3     the improved server comprising:

4         a queryable cache that contains copies of certain of the datasets and is local to the server,  
5     the improved server receiving a query for a remote dataset in a form required by the  
6     interface from the program, determining whether a copy of a dataset to be queried is present  
7     in the queryable cache, and, if the copy is present, querying the copy, and otherwise  
8     querying the remote dataset,  
9     whereby the queryable cache is transparent to the program.

1     **25.** (amended) The improved server set forth in claim 24 wherein  
2         the program uses global identifiers for the remote data sets and  
3         the copies in the queryable cache have local identifiers; and  
4     the improved server further comprises:

5         a query analyzer that receives the global identifier for a dataset being queried and if  
6     there is a copy of the data set indicated by the global identifier, returns the local identifier to  
7     the server,  
8     the server using the local identifier to query the copy.

1     **26.** (amended) The improved server set forth in claim 25 wherein:

2         the query analyzer further indicates to the server whether the copy of the dataset is in  
3     the queryable cache.

1     **27.** (amended) The improved server set forth in claim 24 further comprising:

2 a dataset manager that determines a dataset for which a copy is needed in the cache,  
3 obtains a copy of the remote dataset, and adds the copy to the cache.

1 **28.** (amended) The improved server set forth in claim 27 wherein:

2 the dataset manager further determines a dataset for which a copy is no longer  
3 needed in the cache and removes the copy from the cache.

1 **29.** (amended) The improved server set forth in any of claims 27 or 28 wherein:

2 the dataset manager determines whether to add or remove a dataset by determining a  
3 likelihood that a query will be made to the dataset.

1 **30.** (amended) The improved server set forth in claim 29 wherein the improved server  
2 further comprises:

3 a query log that lists past queries that have been made to the standard interface  
4 and

5 the dataset manager uses the query log to determine a likelihood that a query will  
6 be made to a dataset.

1 **31.** (amended) The improved server set forth in claim 29 wherein:

2 the dataset manager uses information about an event that will result in queries to  
3 a dataset to determine a likelihood that a query will be made to a dataset.

1 **32.** (amended) The improved server set forth in claim 31 wherein:

2 the dataset manager uses information about a time of occurrence of the event to  
3 determine a likelihood that a query will be made to a dataset.

1       **33.** (amended) The improved server set forth in claim 24 wherein

2                   when a change occurs in a remote dataset of the remote datasets, an indication  
3       including the change is sent to the server without intervention by the server and

4                   the improved server further comprises:

5                   an update receiver that receives the indication and modifies any copy of the  
6       changed dataset as required by the indication.

1       **34.** (amended) A method of querying datasets in a server that provides a standard

2                   interface for querying remote data sets to a program executing on the server

3       the method comprising the steps of:

4                   receiving a query for a remote dataset in a form required by the standard  
5       interface;

6                   determining whether a copy of a dataset to be queried is present in a queryable  
7       cache local to the server; and

8                   if the copy is present in the queryable cache, querying the copy and otherwise  
9       querying the remote dataset,

10       whereby the queryable cache is transparent to the program.

1       **35.** The method set forth in claim 34 wherein

2                   the form required by the standard interface uses global identifiers for the remote  
3       data sets and

4                   the copies in the queryable cache have local identifiers; and

5       the method further includes the steps of :

6                   providing the global identifier for a dataset being queried to a query analyzer in  
7       the server; and

8           if there is a copy of the data set indicated by the global identifier, receiving the  
9           local identifier from the query analyzer,  
10          the local identifier being used in the step of querying the local copy.

1          **53.** The method set forth in claim 35 further comprising the step of:  
2           receiving an indication from the query analyzer whether the copy is present in the  
3           queryable cache.

1          **54.** The method set forth in claim 34 further comprising the steps of:  
2           determining a dataset for which a copy is needed in the cache;  
3           obtaining the copy; and  
4           adding the copy to the cache.

1          **55.** The method set forth in claim 54 further comprising the steps of:  
2           determining a dataset for which a copy is no longer needed in the cache; and  
3           removing the copy from the cache.

1          **56.** The method set forth in any one of claims 27 or 28 wherein:  
2           the step of determining a dataset is performed by determining a likelihood that a  
3           query will be made to the dataset.

1          **57.** (amended) The method set forth in claim 56 wherein:  
2           in the step of determining a dataset, a query log that lists past queries is used to  
3           determine the likelihood that a query will be made.

1          **58.** The method set forth in claim 56 wherein:  
2           in the step of determining a dataset, information about an event that will result in  
3           queries to a dataset is used to determine the likelihood that a query will be made.

1   **59.** The method set forth in claim 58 wherein:  
2           in the step of determining a dataset, information about a time of occurrence of the  
3   event is used to determine the likelihood that a query will be made.

1   **60.** The method set forth in claim 34 wherein  
2           when a change occurs in a remote dataset of the remote datasets, an indication  
3   including the change is sent to the server without intervention by the server and  
4   the method further comprises the steps of:  
5           receiving the indication and modifying any copy of the changed dataset as  
6   required by the indication.

1   **84.** A memory device characterized in that:  
2           the memory device contains code which, when executed by a processor, performs a  
3   method of querying datasets in a server that provides a standard interface for querying remote  
4   data sets to a program executing on the server,  
5           the method comprising the steps of  
6           receiving a query for a remote dataset in a form required by the standard  
7   interface;  
8           determining whether a copy of a dataset to be queried is present in a queryable cache  
9   local to the server; and  
10          if the copy is present in the queryable cache, querying the copy and otherwise querying  
11   the remote dataset,  
12   whereby the queryable cache is transparent to the program.

1   **85.** The memory device set forth in claim 84 further characterized in that:  
2           in the method, the form required by the standard interface uses global identifiers  
3   for the remote data sets and

4           the copies in the queryable cache have local identifiers; and  
5           the method further includes the steps of :  
6           providing the global identifier for a dataset being queried to a query analyzer in  
7           the server; and  
8           if there is a copy of the data set indicated by the global identifier, receiving the  
9           local identifier from the query analyzer,  
10          the local identifier being used in the step of querying the local copy.

1   **86.** The memory device set forth in claim 84 wherein the method further comprises the  
2   steps of:  
3           determining a dataset for which a copy is needed in the cache;  
4           obtaining the copy; and  
5           adding the copy to the cache.

1   **87.** The memory device set forth in claim 86 wherein the method further comprises the  
2   steps of:  
3           determining a dataset for which a copy is no longer needed in the cache; and  
4           removing the copy from the cache.

1   **88.** The memory device set forth in any one of claims 86 or 87 wherein:  
2           the step of determining a dataset is performed by determining a likelihood that a  
3   query will be made to the dataset.

1   **89.** (amended) The memory device set forth in claim 88 wherein:  
2           in the step of determining a dataset, a query log that lists past queries is used to  
3   determine the likelihood that a query will be made.

1   **90.** The memory device set forth in claim 88 wherein:

2 in the step of determining a dataset, information about an event that will result in  
3 queries to a dataset is used to determine the likelihood that a query will be made.

1 **91.** The memory device set forth in claim 90 wherein:

2 in the step of determining a dataset, information about a time of occurrence of the  
3 event is used to determine the likelihood that a query will be made.

1 **92.** The memory device set forth in claim 84 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication  
3 including the change is sent to the server without intervention by the server and  
4 the method further comprises the steps of:

5 receiving the indication and modifying any copy of the changed dataset as  
6 required by the indication.

1 **93.** The memory device set forth in claim 85 wherein the method further comprises the  
2 step of:

3 receiving an indication from the query analyzer whether the copy is present in the  
4 queryable cache.

1 **112.** Apparatus for responding to a request, the request including one or more specifiers  
2 referring to objects belonging to a plurality thereof in a distributed database system that  
3 includes a plurality of database systems and

4 the apparatus comprising:

5 a first database system of the plurality; and

6 a redirector which responds to the request when the request includes a specifier  
7 that cannot be interpreted in the first database system by causing the request to be  
8 executed at least in part in a second database system of the plurality, the request  
9 otherwise being executed in the first database system.

1 **113.** The apparatus set forth in claim 112 wherein:

2 the objects in the first database system include copies of objects contained  
3 in at least one other database system belonging to the distributed database system.

1 **114.** The apparatus set forth in claim 113 wherein:

2 the first database system functions as a cache with regard to the objects whose  
3 copies are included therein.

1 **115.** The apparatus set forth in claim 113 wherein the other database system is the  
2 second database system.

1 **116.** The apparatus set forth in claim 115 wherein:

2 the first database system functions as a cache with regard to the second database  
3 system.

1 **117.** The apparatus set forth in any one of claims 112 through 116 wherein:

2 the apparatus is local to a server of the type that provides a program executing in  
3 the server with a standard interface for querying databases; and  
4 the requests include queries received via the standard interface.

1 **118.** The apparatus set forth in claim 117 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 **119.** A method of responding to a request, the request including one or more specifiers  
2 that refer to objects belonging to a plurality thereof in a distributed database system that  
3 includes a plurality of database systems and  
4 the method comprising the steps of:

5 receiving the request in a first database system of the plurality;

6 determining whether the request includes a specifier that cannot be interpreted in  
7 the first database system of the plurality; and



8           when the request includes such a specifier, causing the request to be executed at  
9   least in part in a second database system of the plurality.

1   **120.** The method set forth in claim 119 wherein:

2           the objects in the first database system include copies of objects contained in at  
3   least one other database system belonging to the distributed database system,  
4   whereby the first database system functions as a cache with regard to the objects whose  
5   copies are included therein.

1   **121.** The method set forth in claim 120 wherein:

2           the other database system is the second database system,  
3   whereby the first database system functions as a cache with regard to the second database  
4   system.

1   **122.** The method set forth in any one of claims 119 through 121 wherein:

2           the first database system is local to a server of the type that provides a program  
3   executing in the server with a standard interface for querying databases; and  
4           in the step of receiving the request, the request is received via the standard  
5   interface.

1   **123.** The method set forth in claim 122 wherein:

2           the server obeys the http protocol and the program is a Web application program.

1   **124.** A memory device characterized in that:

2           the memory device contains code which, when executed in a processor, performs  
3   a method of responding to a request, the request including one or more specifiers that  
4   refer to objects belonging to a plurality thereof in a distributed database system that  
5   includes a plurality of database systems and  
6   the method comprising the steps of:

7           receiving the request in a first database system of the plurality;

8           determining whether the request includes a specifier that cannot be interpreted in  
9   the first database system of the plurality; and  
10          when the request includes such a specifier, causing the request to be executed at  
11   least in part in a second database system of the plurality.

1   **125.** Apparatus for caching copies of objects belonging to a subset of the objects  
2   belonging to a first database system that returns an object in response to a request  
3   therefor, the request including one or more specifiers referring to the objects and  
4   the apparatus comprising:

5          a second database system that contains the copies; and  
6          a redirector that responds to the request when the request includes a specifier that  
7   cannot be interpreted in the second database system by causing the request to be executed  
8   at least in part in the first database system, the request otherwise being executed in the  
9   second database system.

1   **126.** The apparatus set forth in claim 125 wherein:

2          the apparatus is local to a server of the type that provides a program executing in  
3   the server with a standard interface for querying databases; and  
4          the requests include queries received via the standard interface.

1   **127.** The apparatus set forth in claim 126 wherein:

2          the server obeys the http protocol and the program is a Web application program.

1   **128.** A method of responding to a request that includes one or more specifiers referring to  
2   objects belonging to a set thereof where the objects are stored in a first database system and  
3   copies of a subset thereof are stored in a second database system,  
4   the method comprising the steps of:

5         receiving the request in the second database system;

6         determining whether the request includes a specifier that cannot be interpreted in  
7   the second database system; and

8         when the request includes such a specifier, causing the request to be executed at  
9   least in part in the first database system instead of in the second database system.

1   **129.** The method set forth in claim 128 wherein:

2         the second database system is local to a server of the type that provides a program  
3   executing in the server with a standard interface for querying databases; and

4         in the step of receiving the request, the request is received via the standard interface.

1   **130.** The method set forth in claim 129 wherein:

2         the server obeys the http protocol and the program is a Web application program.

1   **131.** A memory device characterized in that:

2         the memory device contains code which, when executed in a processor,  
3   performs a method of responding to a request that includes one or more specifiers  
4   referring to objects belonging to a set thereof where the objects are stored in a first  
5   database system and copies of a subset thereof are stored in a second database  
6   system,

7   the method comprising the steps of:

8         receiving the request in the second database system;

9           determining whether the request includes a specifier that cannot be  
10   interpreted in the second database system; and  
11           when the request includes such a specifier, causing the request to be  
12   executed at least in part in the first database system instead of in the second  
13   database system.